"Express Mail" mailing label number EV 164034092 US

Date of Deposit: November 18, 2003

Attorney Docket No.15072US02

SCHEME FOR OPTIMAL SETTINGS FOR DDR INTERFACE

RELATED APPLICATIONS

[01] This application makes reference to, claims priority to, and claims the benefit of United States Provisional Patent Application 60/485,597 (attorney docket number 15072US01) filed on July 8, 2003, entitled "Scheme for Optimal Settings for DDR Interface," the complete subject matter of which is hereby incorporated herein by reference, in its entirety.

FEDERALLY SPONSORED RESEARCH OR DEVELOPMENT

[02] [Not Applicable]

[MICROFICHE/COPYRIGHT REFERENCE]

[03] [Not Applicable]

BACKGROUND OF THE INVENTION

[04] Signal timing is a critical aspect of high-speed digital circuit design. Reading data from memory and writing data to memory can be erroneous if control signals are not in sync with each other. In high frequency digital design, control signals can go out of sync due to different length of tracks they traverse on PCB, physical characteristics of the devices mounted on the board and changes in environment in which circuit is working.

[05] Further limitations and disadvantages of conventional and traditional systems will become apparent to one of skill in the art through comparison of

such systems with the inventions as set forth in the remainder of the present application with reference to the drawings.

BRIEF SUMMARY OF THE INVENTION

[06] Aspects of the present invention may be found in, for example, methods of optimizing a plurality of numerically controlled delay lines (NCDLs) in a DDR memory controller. A method in accordance with the present invention may comprise, for example, one or more of the following: acquiring a plurality of statistics, the plurality of statistics defining an operating region for the DDR memory controller; and calculating optimal values for the plurality of NCDLs, the optimal values calculated using the plurality of statistics.

[07] In another embodiment, there is an article of manufacture comprising a computer readable medium. The computer readable medium stores a plurality of instructions. Execution of the plurality of instructions causes acquiring a plurality of statistics, the plurality of statistics defining an operating region for a DDR memory controller; and calculating optimal values for a plurality of numerically controlled delay lines (NCDLs) in the DDR memory controller, the optimal values calculated using the plurality of statistics.

[08] These and other advantages, aspects and novel features of the present invention, as well as details of an illustrated embodiment thereof, will be more fully understood from the following description and drawings.

BRIEF DESCRIPTION OF SEVERAL VIEWS OF THE DRAWINGS

[09] **Fig. 1A** is a block diagram of a DDR Memory Controller, in accordance with an embodiment of the present invention;

[10] **Fig. 1B** is a signal plot of a center aligned DQS signal with respect to a data signal, in accordance with an embodiment of the present invention;

[11] **Fig. 2** is a blocked diagram of a delay locked loop, in accordance with an embodiment of the present invention;

[12] **Fig. 3** is a flowchart illustrating an embodiment of a method for optimizing a plurality of numerically controlled delay lines (NCDLs) in a DDR memory controller, in accordance with an embodiment of the present invention;

[13] **Fig. 4** is a plot of a passing count versus gate NCDL, in accordance with an embodiment of the present invention; and

[14] **Fig. 5** is an exemplary hardware environment, wherein the present invention may be practiced.

DETAILED DESCRIPTION OF THE INVENTION

[15] Referring now to **Fig. 1A**, there is illustrated a DDR memory controller in accordance with an embodiment of the present invention. The DDR memory controller 100 comprises a master delay locked loop (DLL) 101, a read numerically controlled delay line (NCDL) 111, a write NCDL 113, a gate logic 114, a gate NCDL 115, two's complimentary adders 103 and 105, a read NCDL offset 107, a write NCDL offset 109, and an internal clock 117.

[16] A DDR-SDRAM (DDR device) is a Double Data Rate Synchronous Dynamic Random Access Memory, which receives and transfers data at both edges of the clock in order to achieve high bandwidth. In order to ensure that the data is received and transferred reliably, DDR defines a bi-directional signal called DQS (or a data strobe signal), and the timing of the data is specified with respect to the edges of this signal.

[17] The DQS is center aligned with respect to data during a write operation to the DDR device. Referring now to **Fig. 1B**, there is illustrated a signal plot of a center aligned DQS signal with respect to a data signal. The DDR device outputs DQS that is edge aligned with respect to data during the read operation. The DDR controller 100 is an agent that interfaces with the DDR device and all the components on the board access the DDR device through the DDR controller 100. During a write operation, the DDR controller is expected to center align (90-degree phase shift) the DQS with respect to the data signal, as illustrated on **Fig. 1B**. DQS signal is tri-stated when there is no read or write operation. The noise can trigger a false read. Hence, during a read operation, the controller first validates the DQS by opening the gate signal. Referring again to **Fig. 1A**, the DQS read signal 140 is received with the incoming data read signal 141. The DQS read signal 140 is then validated by the controller 100 by opening the gate 114. The controller 100 then phase shifts the DQS read signal 140 by 90 degree to reliably register the incoming data. Since the incoming DQS 140 will not be in phase with the controller's internal clock 117, gate opening is achieved by delaying the internal clock generated gate signal with the use of the gate NCDL 115.

[18] A numerically controlled delay line (NCDL) is a piece of hardware that delays the signal passed through it. The amount of delay is proportional to a value set at the phase control input of the NCDL. DLL (Delay locked loop) is a piece of hardware that uses similar NCDLs to phase lock the input and output clock. Referring now to **Fig. 2**, there is illustrated a DLL, using two NCDLs, 201 and 203, to lock the output at 180 degree with respect to the input clock 205. Since the value of the up/down counter 207 is fed to both NCDL 201 and NCDL 203, and the DLL achieves a phase lock of 180 degrees, the counter 207 contains a numerical value that would give a 90 degrees phase shift between the input and the output of any other similar NCDL.

[19] Referring again to **Fig. 1A,** the phase shift on the DQS signals 130 and 140 is achieved by using an NCDL, whose numerical input is generated by a DLL. This setup ensures that the phase shift obtained tracks Process-Voltage-Temperature (PVT) variation. However, because of board skews, the phase shift obtained from the DLL might not exactly produce a 90 degrees phase shift on the DQS path. An embodiment of the present invention allows for an offset of the numerical value produced by the DLL in order to achieve optimum setting for the DDR controller 100. It also allows for the gate opening 14 to be fixed at an optimal point allowing reliable operation across PVT variations.

[20] The DDR controller 100 has three NCDLs – a gate NCDL 115, a read NCDL 111, and a write NCDL 113. The read NCDL 111 is used to phase shift the DQS read signal 140 with respect to the data read signal 141 when data is read from the DDR device. The write NCDL 113 is used to phase shift the DQS write signal 130 with respect to the data write signal 131 when data is written to the DDR device connected to the DDR controller 100. The gate NCDL 115 allows for an optimization of opening the gate logic 114 for the incoming DQS read signal 140 during a read operation.

[21] The master DLL 101 outputs a number that, when programmed in an NCDL, would produce a 90-degree phase shift in the signal passing through it. Since similar NCDLs are being used as the read NCDL 111 and the write NCDL 113, the numerical value from the master DLL 101 produces the same 90-degree phase shift

for the read NCDL 111 and the write NCDL 113. However, to compensate for the board skews, a programmable offset is added to the numerical output of the master DLL 101. The read and write NCDLs have separate programmable offset registers, providing a read NCDL offset value 107 and a write NCDL offset value 109. The output from the master DLL 101 and the two offset values, 107 and 109, are fed into compliment adders 103 and 105. The values that get programmed into the read NCDL 111 and the write NCDL 113 are the two's compliment additions of the DLL output value and the respective NCDL offsets 107 and 109. However, the gate NCDL 115, that is used to delay signal entering the gate logic 114, is programmed with an absolute value.

[22] In accordance with an embodiment of the present invention, a software program tests all the possible combinations of the write NCDL offset 109, the read NCDL offset 107, and the gate NCDL 115 under stressful condition. The software then programs NCDL registers of the DDR memory controller 100 optimally, bringing sync relationship of 90-degree phase-shift between a DQS signal and a data signal. Even though offset is programmed into the read and write NCDLs, here onwards these programming values will be referred to as read NCDL and write NCDL. The range of NCDL values, for which the DDR memory controller 100 works reliably defines an operating region for the DDR memory controller. The optimal working point may then be calculated using the operating region.

[23] Referring now to **Fig. 3**, there is illustrated a flow diagram for optimizing a plurality of numerically controlled delay lines (NCDLs) in a DDR memory controller, in accordance with an embodiment of the present invention. The method 300 comprises acquiring statistics that define the operating region of the DDR memory controller and finding out the optimal NCDL values for the read NCDL, the write NCDL, and the gate NCDL from the acquired statistics.

[24] At 301, the necessary statistics are acquired by calculating the working range of the read NCDL offset and the write NCDL offset, as well as a passing count for each of the gate NCDL values. Passing count is the number of working combinations of read and write NCDL values for a particular gate NCDL value. The

stressful condition, under which these statistics are obtained is created by running SSO (Simultaneously Switched Outputs) test multiple times for each combination of the NCDL values. In the SSO test, all the lines in a data bus are switched simultaneously from 0 to 1, or from 1 to 0. This stresses the power supply resulting in more slanting edges and reduced data eye width. Each SSO test is preceded by one random pattern write and read back test. This is necessary to eliminate false passing of the SSO test – case in which write fails but read still passes due to correct write in the previous SSO test. For each of the NCDL combination, SSO test is run multiple times at different critical memory locations (e.g. different bank boundaries etc) throughout the available memory. This has the effect of reducing data eye width.

[25] The statistics acquired during 301 are used to calculate the optimal setting for the DDR controller. First from the statistics, corner gate value is calculated during 303. Referring now to **Fig. 4**, there is illustrated a plot of passing count versus gate NCDL values. The corner gate value is defined as the gate value that has 90% of the maximum number of passing counts.

[26] Referring again to **Fig.3**, MasterDLL(avg), MasterDLL(0-tap), MasterDLL(per tap) and Gate(per tap) values are calculated during 305. The following formulas may be used for the calculations during 305:

MasterDLL(avg) = Average value of high and low master DLL value

MasterDLL(0-tap) = Amount of 0-tap delay in picosecond

MasterDLL(per-tap) = Per tap delay of master DLL in picosecond

Gate(per-tap) = Per tap delay of Gate NCDL in picosecond

[27] At 307, a factor value X is calculated, in accordance with the following formula:

X = Factor defined as ((clk_period/4) – tdqsck)/(clk_period/4), where

tdqsck = the clock to DQS skew as defined by the DDR datasheet.

This factor value gives the effective 90-degree delay tap value of the gate NCDL.

[28] At 309, the final gate NCDL value is calculated, which represents the optimum value for the gate NCDL. The following formula is utilized:

Final Gate NCDL = X * {MasterDLL(avg) + MasterDLL(0-tap)/Master(per- tap)} * Master(per-tap)/Gate(per-tap) + GateCorner

[29] The optimum value for the read NCDL, the Final Read NCDL, is calculated at 311, by averaging all Read NCDL values in the Read NCDL range for the Final Gate NCDL value obtained during 309.

[30] The optimum value for the write NCDL, the Final Write NCDL, is calculated during 313, by averaging all Write NCDL values in Write NCDL range for the Final Gate NCDL value obtained during 309.

[31] If corner gate value is not found in the statistics, then the statistics may be divided in two parts. One part will correspond to gate values having passing counts more than 90% of the maximum number of passing counts. The second part will correspond to gate values having passing counts less than 90% of the maximum number of passing counts. The second part may then be discarded and optimum NCDL values may be calculated from the first part by averaging, as follows:

Final Gate NCDL = Average of all Gate NCDL values

Final Read NCDL = Average of all Read NCDL values in all Read NCDL ranges for all Gate NCDLs

Final Write NCDL = Average of all Write NCDL values in all Write NCDL ranges for all Gate NCDLs

[32] Referring now to **Fig. 5**, a representative hardware environment for a computer system 58 for practicing the present invention is depicted. A CPU 60 is interconnected via system bus 62 to random access memory (RAM) 64, read only memory (ROM) 66, an input/output (I/O) adapter 68, a user interface adapter 72, a communications adapter 84, and a display adapter 86. The input/output (I/O) adapter 68 connects peripheral devices such as hard disc drives 40, floppy disc drives 41 for reading removable floppy discs 42, and optical disc drives 43 for

reading removable optical disc 44 (such as a compact disc or a digital versatile disc) to the bus 62. The user interface adapter 72 connects devices such as a keyboard 74, a mouse 76 having a plurality of buttons 67, a speaker 78, a microphone 82, and/or other user interfaces devices such as a touch screen device (not shown) to the bus 62. The communications adapter 84 connects the computer system to a data processing network 92. The display adapter 86 connects a monitor 88 to the bus 62.

[33] An embodiment of the present invention can be implemented as a file resident in the random access memory 64 of one or more computer systems 58 configured generally as described in **Fig. 5**. Until required by the computer system 58, the file may be stored in another computer readable memory, for example in a hard disc drive 40, or in removable memory such as an optical disc 44 for eventual use in an optical disc drive 43, or a floppy disc 42 for eventual use in a floppy disc drive 41. The file can contain a plurality of instructions executable by the computer system, causing the computer system to perform various tasks, such effectuating the flow chart described in **Fig. 3**.

[34] One skilled in the art would appreciate that the physical storage of the sets of instructions physically changes the medium upon which it is stored electrically, magnetically, or chemically so that the medium carries computer readable information.

[35] While the invention has been described with reference to certain embodiments, it will be understood by those skilled in the art that various changes may be made and equivalents may be substituted without departing from the scope of the invention. In addition, many modifications may be made to adapt particular situation or material to the teachings of the invention without departing from its scope. Therefore, it is intended that the invention not be limited to the particular embodiment(s) disclosed, but that the invention will include all embodiments falling within the scope of the appended claims.